

LU04b - Statische Code Analyse

Hilfsmittel der IDE

WebStorm ist eine integrierte Entwicklungsumgebung (IDE) von JetBrains, die speziell für Webentwickler entwickelt wurde. Sie bietet eine Reihe von Funktionen, um Ihnen zu helfen, einen konsistenten und sauberen Code-Stil einzuhalten:

Automatische Formatierung

WebStorm kann Ihren Code automatisch formatieren, um die Einrückung, Abstände und Zeilenumbrüche gemäss den konfigurierten Code-Style-Regeln anzupassen. Sie können die automatische Formatierung verwenden, indem Sie Strg + Alt + L (Windows/Linux) oder Cmd + Alt + L (Mac) drücken.

Code-Style-Konfiguration

WebStorm ermöglicht Ihnen, benutzerdefinierte Code-Style-Regeln für verschiedene Sprachen wie HTML, CSS und JavaScript einzustellen. Sie finden die Einstellungen unter **File > Settings > Editor > Code Style**. Sie können die Regeln für Einrückung, Leerzeichen, Umbrüche und Leerzeilen anpassen.

Linters und Code-Analyse-Tools

WebStorm unterstützt die Integration von Lintern wie z.B TSLint. Diese Tools können Ihnen dabei helfen, Ihren Code-Stil zu überprüfen und sicherzustellen, dass er den festgelegten Standards entspricht.

Live-Vorlagen und Code-Snippets

Sie können Live-Vorlagen und Code-Snippets verwenden, um häufig verwendete Codeblöcke schnell und konsistent einzufügen. Sie können eigene Vorlagen erstellen oder die von WebStorm bereitgestellten Vorlagen verwenden.

Datei- und Code-Vorlagen

WebStorm bietet Datei- und Code-Vorlagen, die als Ausgangspunkt für neue Dateien dienen und den vom Projekt vorgegebenen Stil beibehalten. Sie können unter **File > Settings > Editor > File and Code Templates** angepasst und verwaltet werden.

Code-Inspektionen

WebStorm führt Code-Inspektionen durch, um mögliche Probleme und Inkonsistenzen im Code-Stil zu erkennen. Die IDE hebt problematische Bereiche hervor und bietet schnelle Lösungen, um den Code-Stil zu verbessern.

Echtzeit-Feedback

WebStorm bietet Echtzeit-Feedback zu Ihrem Code-Stil, während Sie tippen. Die IDE kann Ihnen dabei helfen, sofortige Korrekturen vorzunehmen und so einen konsistenten Stil beizubehalten.

Indem Sie diese Funktionen von WebStorm nutzen, können Sie sicherstellen, dass Ihr Code konsistent, sauber und in Übereinstimmung mit den gewünschten Code-Style-Regeln ist.

Linter für HTML installieren

Ein Linter ist ein Programm, das den Quellcode analysiert und auf Probleme, Inkonsistenzen, Verstöße gegen Best Practices oder Stilrichtlinien hinweist. Linter helfen Entwicklern, potenzielle Fehler, Stilprobleme und schlechte Praktiken in ihrem Code frühzeitig zu erkennen, bevor sie zu Problemen führen. Sie unterstützen dabei, sauberer und konsistenten Code zu schreiben und die Codequalität insgesamt zu verbessern. Wenn Sie Ihren Code auf Github pushen, wird dieser durch [htmlhint](#) einer statischen Code-Analyse unterzogen. Damit Sie diese auch bereits auf Ihrem Notebook machen können installieren Sie htmlhint auf Ihrem Notebook.

Installation

HTMLHint ist ein Linter für HTML-Code, der auf Node.js basiert. Er hilft Ihnen, sicherzustellen, dass Ihr HTML-Code konsistent und gut formatiert ist. Um HTMLHint auf einem Windows-System zu installieren, führen Sie die folgenden Schritte aus:

1. Stellen Sie sicher, dass Node.js auf Ihrem System installiert ist. Falls Node.js noch nicht auf Ihrem Computer installiert ist, können Sie es von der offiziellen Website (<https://nodejs.org/>) herunterladen und installieren. Wählen Sie die empfohlene LTS-Version (Long Term Support) für maximale Stabilität.
2. Öffnen Sie die Windows-Eingabeaufforderung (CMD) oder PowerShell als Administrator.
3. Installieren Sie HTMLHint global, indem Sie den folgenden Befehl eingeben:
`npm install -g htmlhint`
4. Nach der Installation können Sie HTMLHint verwenden, um Ihre HTML-Dateien zu überprüfen. Um beispielsweise eine Datei namens index.html zu überprüfen, navigieren Sie zum Verzeichnis, in dem sich die Datei befindet, und führen Sie den folgenden Befehl aus:
`htmlhint index.html`

HTMLHint wird die Datei analysieren und Probleme oder Verstöße gegen die Standardregeln melden. Sie können die Regeln anpassen, indem Sie eine Konfigurationsdatei namens `.htmlhintrc` in Ihrem Projektverzeichnis erstellen und die gewünschten Regeln darin festlegen. Um Ihre Konfiguration zu

verwenden, führen Sie diesen Befehl aus:

```
htmlhint --config .\github\linters\.htmlhintrc index.html
```

Natürlich müssen Sie den Namen der zu prüfenden Datei (hier `index.html`) anpassen.

Um alle HTML-Dateien im aktuellen Projekt zu prüfen, verwenden Sie:

```
htmlhint --config .\github\linters\.htmlhintrc *.html
```

Informationen zu den verfügbaren Regeln und ihrer Konfiguration finden Sie in der HTMLHint-Dokumentation (<https://htmlhint.com/docs/user-guide/list-rules>).

Webstorm Konfigurieren



Starten Sie Webstorm neu, nachdem Sie NodeJS und htmlhint installiert haben.

Sie müssen Webstorm die Benutzung von Powershell erlauben. Fügen Sie dazu die Anweisung ins Terminal von Webstorm ein und führen sie die Anweisung aus.

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted -Force
```

Benutzen

[Öffnen Sie die das Terminal in Webstorm.](#)



Dieses Terminal öffnet sich automatisch im Workspace-Ordner. Tippen Sie nun `htmlhint --config .\github\linters\.htmlhintrc` ins Terminal-Fenster und hmtlhint wird ihren Code statisch analysieren und ihnen einen Bericht liefern.



Da wir eine an die BZZ-Standards angepasste htmlhint-Config verwenden müssen **wir den Pfad zur Config immer angeben**.

Last update:

2025/05/15 modul:m293:learningunits:lu04:statischecodeanalyse <https://wiki.bzz.ch/modul/m293/learningunits/lu04/statischecodeanalyse>
13:12

Windows: htmlhint --config
.github\linters\.htmlhintrc



Unix: htmlhint --config
.github/linters/.htmlhintrc

```
Terminal Local + ~
kevinmaurizi@Kevins-MBP M293_LU04_01_Codingstyle % htmlhint
Config loaded: /Users/kevinmaurizi/WebstormProjects/M293_LU04_01_Codingstyle/.htmlhintrc
/Users/kevinmaurizi/WebstormProjects/M293_LU04_01_Codingstyle/gleitschirm.html
L4 |<META CHARSET="UTF-8">
^ The html element name of [ META ] must be in lowercase. (tagname-lowercase)
L4 |<META CHARSET="UTF-8">
^ The attribute name of [ CHARSET ] must be in lowercase. (attr-lowercase)
L5 |<TITLE>Gleitschirmfliegen - Tips und Ausrüstung</title>
^ The html element name of [ TITLE ] must be in lowercase. (tagname-lowercase)
L7 |<BODY>
^ The html element name of [ BODY ] must be in lowercase. (tagname-lowercase)
L8 |<H1>Gleitschirmfliegen - Einsteigertips und Ausrüstung</h1><h2>Einführung ins Gleitschirmfliegen</h2...
^ The html element name of [ H1 ] must be in lowercase. (tagname-lowercase)
L8 |...Ausrüstung, um sicher fliegen zu können.</P>
^ The html element name of [ P ] must be in lowercase. (tagname-lowercase)
L9 |<H2>Ausrüstung</h2><p>Um Gleitschirmfliegen zu können, benötigen Sie eine Grundausstattung. Diese umf...
^ The html element name of [ H2 ] must be in lowercase. (tagname-lowercase)
L9 |..., einer Funkanlage und eines GPS-Geräts.</P>
^ The html element name of [ P ] must be in lowercase. (tagname-lowercase)
L10 |<H2>Tipps für Anfänger</h2><p>1. Wählen Sie einen geeigneten Startplatz</p><p>2. Über Sie das Starten...
^ The html element name of [ H2 ] must be in lowercase. (tagname-lowercase)
L11 |<H3>Beliebte Fluggebiete</h3>
^ The html element name of [ H3 ] must be in lowercase. (tagname-lowercase)
```



Passen Sie nun den Code an, bis ihr Code keine Fehler mehr enthält.

[m293-LU04](#), [m293-H2G](#), [m293-H2F](#), [m293-H2E](#)



© Marcel Suter, Kevin Maurizi

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/modul/m293/learningunits/lu04/statischecodeanalyse>

Last update: **2025/05/15 13:12**

